Journal of Network and Innovative Computing
ISSN 2160-2174 Volume 3 (2015) pp. 115-127

The Impact of Knowledge Management Processes in Pair Programming Practice

Mazida Ahmad[1], Ariffin Abdul M[2], Mazni Omar[3], Azman Yasin[4], Rohaida Romli[5]

[1345] School of Computing, College of Arts and Sciences
Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia
mazida@uum.edu.my; s814049@student.uum.edu.my; {mazni, yazman, aida}@uum.edu.my; anasyafiqah91@gmail.com

[2] School of Multimedia Technology and Communication, College of Arts and Sciences
Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia
am.ariffin@uum.edu.my

***Abstract*: One of Extreme Programming practices is Pair Programming (PP) (the pair consists of a driver and a navigator), which is used for promoting knowledge sharing among students. This practice encourages students to think creatively on programming solutions, and simplify learning, especially for difficult courses such as Java. By applying PP, students are enforced to improve their social skills while communicating in the pair. Among the numerous benefits of PP, statistics show that knowledge sharing, communication, and transfer between the driver and the navigator can improve the code quality. Therefore, this study aims at proposing a conceptual model of a PP knowledge-based sharing for improving programming skills. In order to achieve the stated aim, PP-based laboratory assignments were conducted and the outcomes were compared to evaluate the impact of PP on code quality produced by participating students with and without adopting the conceptual model. The conceptual model has been validated by analyzing the collected data from the participants of PP-based laboratory assignment using Partial Least Square form of Structural Equation Modeling (PLS-SEM). In the end, this study found that socialization, combination, and internalization are the determinant factors for achieving better code quality in PP environment. The findings of this study would be benefiting the academic environment especially the Agile programmers in the PP domain.**

Keywords: Pair programming, Tacit knowledge, Code quality and SECI model**.**

I. Introduction

The success of Pair Programming (PP) in IT industry has been seen in terms of enhancing knowledge transfer [1], facilitating integration of novice members [2], reducing costs for training [2], and improving coding structure [3]. This encourages practitioners in pedagogical context to rely on PP to overcome students' failure in programming course. Additionally, it not only encourages students to accept programming curricula, but also encourages innovation in producing better end-programs [1]. Consequently, Software Engineering (SE) community has accepted the PP as one of many innovative approaches that has been considered to overcome distortions in programming skills in Computer Science (CS) and SE courses [4][5]. Eventually, in the late 1990s, PP has been embedded in the teaching of CS [6].

Improving the programming skills of the students in higher learning institutions takes the much concern of this study. This is because good code quality is an indicator to good programming skills. Coming up with good code quality requires a sufficient amount of a student's personal knowledge. On that basis, the idea of constructing a conceptual model that can improve the programming skills among students of higher learning institutions has been initiated, which has been set as the main aim of this study.

Constructing the conceptual model requires this study to satisfy two needs, in which the first is to employ a well-known model that deals with knowledge management and impact on individuals' personal knowledge (tacit knowledge). Meanwhile the second need is to use a practice that is reliable in CS and SE community, deals with knowledge management, and fosters tacit knowledge. For the first need, this study employs the model by Nonaka and Takeuchi [7], which is socialization-externalization-combination-internalization (SECI), known as "knowledge creation theory". The model has been decided for consideration because it is not only a knowledge management component; it also builds up interaction for knowledge transfer [8]. Meanwhile, for the second need, this study employs PP.

II. Review

A. First Lab Procedure

PP is a collaborative programming manner of Extreme Programming practices of Agile software development family. In software industry, PP has been widely practiced for programming solution, where two programmers working side by side on one computer on the same problem with great success [9]. The element that distinguishes PP from other collaborative programming styles is the terms: "driver", "navigator" and the technique they adapt to process a task [10]. Procedurally, the pair use one set of workstation in solving the problem. The pair is imposed to design, code, diagnose, and develop a project [11]. In the practice, both programmers enthusiastically interact in the pair utilizing role-base procedure [12]. The driver is one of two PP partners who code for solving the problem [1][13][14] while the navigator observes driver's job while on the keyboard. However he/she has strategic duties; brainstorms the whole structure, focusing on tactical errors, and feeding the coding with proper alternatives [1][14]. Although the navigator may sit for a long time and say nothing, just observing the codes and coding process, especially when the driver is proceeding well, it does not give bad impression or any misunderstanding on the on-going activity. Both partners must remain vigilant and ready to guide the driver and pick errors up along the work [10][11]. When necessary, they switch their roles to improve their work and learn appropriate skills [11].

B. Knowledge Sharing

Generally, knowledge creation refers to the ability to construct information and arranged data [15]. In 1958, Michael Polanyi [16] has ignited the classifications of knowledge as tacit and explicit. Tacit knowledge is characterized by [17] as the own experience and expertise of a person that is hard to be described and understood by others. In addition, it is classified as the ability in doing calculation and making decision. It is therefore an applied knowledge, which a person gains in doing a daily job instead of through official instructions. This agrees with Kavitha and Ahmed [18], who previously addressed that tacit knowledge preserves in individual's mind in the mode of experience, memory, skills, inventiveness, and resourcefulness. This means that tacit knowledge is a resultant of an individual's experience stored in mind, which is not easy to be formalized even not to be measured facilely and is very context-specific [19]. Factors influencing tacit knowledge includes everything that the person has mentally ratified in the learning phase [20]. Besides not easy to express, it is also hard to transfer due to the differences in formulation of speech and understanding [21][22] and is difficult to retain [23].

In contrast, explicit knowledge can be transformed into a form of words, email, data [24] related to tangible resources. It is supported by archived information such as curricula [25], documented experience [25], and books in addition to web (could be a source of tacit knowledge) [26]. This means that explicit knowledge is easy to explain [27], copy [27], and capture [28], and can be divulged easily [19].

On the other hand, knowledge management has been defined by various scholars in different ways. It is seen as planning, controlling, organizing, and inspiring individuals, systems, as well as processes in an establishment so as to enhance knowledge asset and utilize it effectively [29][30]. To some extent, it is perceived as a procedure in organizing knowledge assets in order to achieve learning in the organization [31]. Based on the previous studies, knowledge management can be defined as a technique in acquiring, converting, and applying knowledge.

As the foundation, knowledge creation as well as the transformation of tacit knowledge into explicit knowledge is recognized as the basic element of knowledge management [32][33]. It is agreed by Crawford et al. [34], who found that it is achieved through person to person or cluster to cluster interaction.

Besides, knowledge management techniques, which are ill-defined with agile methodological procedures, have been the rationale towards agile practice recognition together with software development and project. This has resulted in a wide recognition of software methodologies in various communities. According to Sharma [35] and Singh and friends [36], such agile techniques include PP, onsite customer, and scrum meetings, also enhance knowledge creation, retention, as well as knowledge dissemination. Commonly, within any organization, individuals are treated with many activities concerning knowledge involving acquiring, using, sharing in addition to sorting knowledge [37].

C. Pair Programming in Knowledge Sharing

Knowledge sharing, as maintained by agilest, is an answer to the current challenges and popular difficulties of software development. It is the main part of knowledge management and a critical mission in Agile [37]. According to Fengjie and friends [38], the knowledge sharing process involves two main parties namely the contributor and the receiver. In the process, the contributor begins by transmitting part of his/her knowledge to the receiver. The receiver will receive the knowledge and try to add his/her understanding and formulates it into his/her knowledge. This scenario is similar with the PP practice, in which the navigator plays the role of the contributor and the driver is the receiver.

In PP practice, knowledge sharing involves social interaction, sharing, and constructing knowledge between the partners. In this scenario, the SECI model is applicable to promote sharing and constructing tacit knowledge between partners in generating codes with high quality. Code quality is an indicator for less number of defects in syntax and it measures the acceptance level of a program among users in terms of reliability, usability, maintainability, and portability [39]. Besides, the literatures agree that expert opinion, effectiveness, academic performance, and number of successful test cases also measure code quality [40].

SECI modeling also facilitates the understanding of the association of interaction and transaction between both tacit and explicit knowledge [8], [41]. Further, Ikujiro and Takeuchi [42] detailed out the four stages. Technically, socialization refers to a state in which tacit knowledge is generated as a result from sharing mental thinking and practical experience during social interaction like informal session, debate, and co-existence [43]. Externalization concerns in articulation of tacit knowledge into documents form which can be later shared with the others, based on the new codified form or explicit knowledge. Hence, externalization phase is meant by 'tacit-explicit' knowledge [19]. Meanwhile, combination, which is denoted by explicit-explicit refers to supporting explicit knowledge with systematic resources in order to uplift the level of unsystematic explicit knowledge [19]. Eventually, the fourth phase of SECI cycle is internalization, in which a systematic explicit knowledge converts to a richer, consistent, and more complicated tacit knowledge (saved in head) [44].

III. Methodology

The study began with defining the research context (in which the research procedure is illustrated in Figure1). Then, the model was built by focusing on a preliminary study that led to the understanding of the SECI model, program quality, and PP. Then, model validation began as an experimental process, which was adapted from the general experimentation in SE [45].

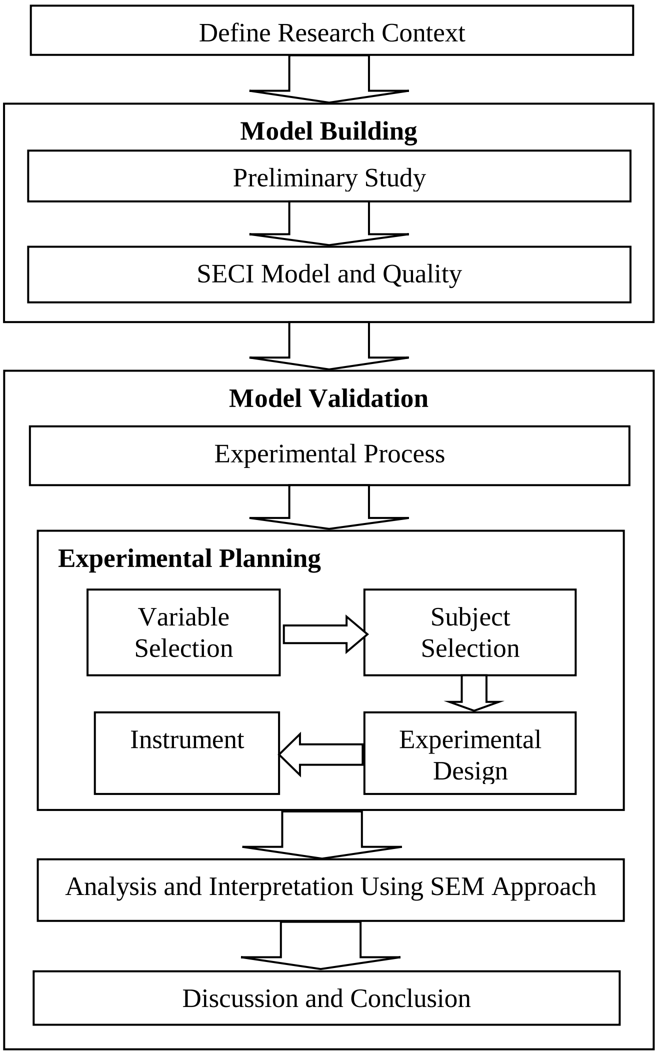

Figure 1. Research procedure

In experiment planning stage, several activities were included in. All variables were specified before conducting the experiment. It was helpful in overcoming the validity threats [46].

In order to investigate the relationships between knowledge sharing and program quality in PP practice, the hypotheses in Table 1 have been formulated. Further, Figure 2 outlines the dependent and independent variables.

Table 1. Hypotheses

No	Hypothesis	Codification	Description
1	H1	S $\xrightarrow{NS}$ CQ	The Socialization process contributes positively to student's code quality without employing SECI process.
2	H2	E $\xrightarrow{NS}$ CQ	The Externalization process contributes positively to student's code quality without employing SECI process.
3	H3	C $\xrightarrow{NS}$ CQ	The Combination process contributes positively to student's code quality without employing SECI process.
4	H4	I $\xrightarrow{NS}$ CQ	The Internalization process contributes positively to student's code quality without employing SECI process.
5	H5	S $\xrightarrow{YS}$ CQ	The Socialization process contributes positively to student's code quality with employing SECI process.
6	H6	E $\xrightarrow{YS}$ CQ	The Externalization process contributes positively to student's code quality with employing SECI process.
7	H7	C $\xrightarrow{YS}$ CQ	The Combination process contributes positively to student's code quality with employing SECI process.
8	N8	I $\xrightarrow{YS}$ CQ	The Internalization process contributes positively to student's code quality with employing SECI process.
9	H9	SECI $\rightarrow$ CQ	SECI process contributes positively to student's code quality.

To evaluate the quality of codes with SECI model, the independent variables undergo the experimentation process. Meanwhile the dependent variables refer to the effects to be measured. In such context, the dependent variables are code quality and elements in the SECI model, which are Socialization, Externalization, Combination and Internalization are the independent variables.

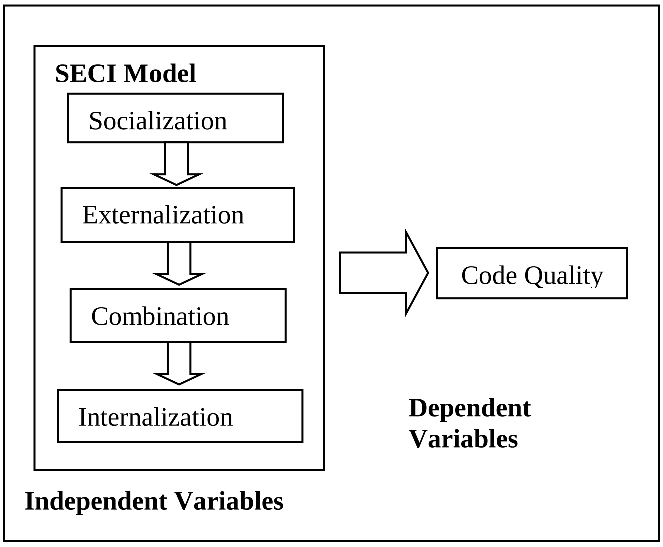

Figure 2. Research process

The subjects of the study are undergraduate students of Collage of Arts and Sciences (CAS) at Universiti Utara Malaysia (UUM). The learning zone (UUM's learning management system) has been used to announce the call for participation. This ensured that all participants involved in the study on voluntary basis. They were motivated by special mark for their curriculum activity called SIRA. Consequently, 108 students participated in the study, in which the condition is that they have been familiar with the fundamentals of Java programming.

They were required to solve two Java programming assignments, assigned by the lecturer. They reflect students' performance in PP practices through pre and post applying SECI phases. Additionally, the participants were required to answer a set of questionnaire that reflects their perception on knowledge sharing between pair programmers in presence of SECI model.

The experiment design of this study concerns on testing the knowledge sharing through applying SECI model and its relationship with the quality of the end-program. In addition, this study has an intention in the manipulation of variables. Hence, the decision in conducting experiment using repeated measures were taken. This makes every student involved in different situations in the experiments.

Two conditions of programming practices were included in this study, which are 1) PP without applying SECI for knowledge transfer in cases of with rotation (denoted by NSYR) and 2) PP with SECI model in cases of with rotation (denoted by YSYR). In short, the repeated measure design is illustrated in Table 2.

Table 2. Repeated Measure Design

V / IV	**Dependent Variable (Code Quality)**	
Independent Variable (SECI-CQ)	NSYR Student Group Socialization Externalization Combination Internalization	YSYR Student Group Socialization Externalization Combination Internalization

Table 2 explains that every student pair applied four different programming practices at different times. This ensures the reliability of the gathered results. The design is further detailed in Figure 3, which visualizes that the first and the second lab experimentations were concerned with PP practice with the absence of SECI implication for knowledge sharing. Meanwhile the third and fourth lab experimentations were incorporated with SECI model. Identical questionnaire was distributed to the participants to measure their level of knowledge during lab activities.

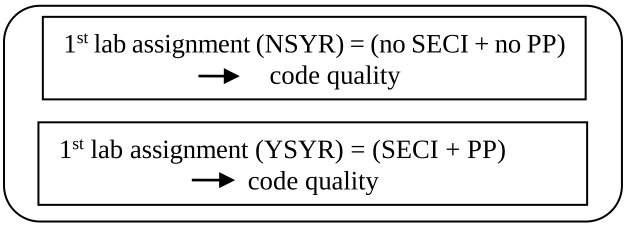

Figure 3. Lab experiments

For the purpose of conducting effective lab experiments and to reduce the effects of biasness, several procedures were taken as detailed out in the next sections. Further, to ensure proper PP sessions in terms of interaction and collaboration, the roles of the instructor and the participants were specified in the following guidelines:

Roles of instructor:
1. Brief students on PP and its practices.
2. Give students chance to choose their adequate pair programmer.
3. Support novice participants with tips in case of difficulty to encourage them to proceed well in completing the task.
4. Explain the problems to the participants in some ways without highlighting the answer except for novice participants who could be supported with tips especially in the early stages.
5. Trace the deployment equality in participation between the pair programmers.

Roles of the participants:
1. Free to choose their adequate pair partner.
2. Ask the instructor for guidance in case of necessary.
3. Discuss with the partner to come out with proper results. Switch the roles (in pairs) as scheduled

A. *First Lab Procedure*

The second lab (the NSYR) was also given one hour, but with roles rotation between the members of the pairs. This enables each member to be a driver for half an hour, while as a navigator for another half an hour. Similarly, the set of questionnaire in the first lab was distributed in this second lab too.

Meanwhile, the third and fourth labs were conducted to investigate the quality of the program in the presence of SECI model in PP practice. The equality in terms of the level of difficulty of the assignments in the four lab sessions was highly ensured. For better implementation of SECI processes, the participants were instructed with a set of guidelines (Table 3) before conducting the third lab. This might positively affect the knowledge sharing between the diver and the navigator of the pairs, and accordingly might impact the final program coding.

Table 3. SECI Guidelines

SECI Stage	**Guideline**

SECI Stage	Guideline
Socialization	Each participant has to think for the solution (in the form of a program) deeply.
Externalization	The members of the pairs need to share by writing a draft code of the program.
Combination	The participants can refer to the Internet, software book, or any source to support their program.
Internalization	Once participants are satisfied with the output code, they can write and run it using the provided computer.

B. Second Lab Procedure

Similarly, one hour was assigned for the fourth lab (YSYR). The aim was to investigate the quality of the program with the presence of SECI, and with pair rotation as illustrated in Figure 3. The members of the pairs were required to switch the roles as a driver and a navigator after the first 30 minutes.

To meet the research objectives, quantitative analysis was used. For the purpose of testing the determined hypotheses, Structural Equation Modeling (SEM), which is an analytical technique involves measurement errors to understand the influencing indicators [47] was run. Also SEM was used to examine whether the conceptual model fits with the collected data through the experiments.

In this study, Partial Least Square (PLS) is employed, utilizing SmartPLS 2.0 as the tool. This is because PLS can be used to avert the limitations of co-variance-based SEM with regards to distributional properties, measurement level, sample size, model complexity, identification, and factor interdependencies [48]. Urbach and Ahlemann [49] stated the criteria for choosing PLS, i.e. PLS makes fewer demands regarding the sample size than other methods.

IV. Findings

Based on the analyzed data using the International Business Management (IBM) Statistical Package for Social Sciences (SPSS) version 20 and the SmartPLS 2.0 tools, findings are discussed in the following subsections.

A. Descriptive Statistics of Respondents for Experiments 1 and 2 (NSYR & YSYR)

The statistical frequency distribution of variables in the questionnaire was classified and presented in a way to reflect the originality of this study. NSYR is a denotation to the experiment in which PP session was conducted without applying SECI process but with role rotation in the pairs. Meanwhile, YSYR denotes the experiment with the incorporation of SECI and pair rotation.

Based on that, the descriptive analytical tables for experiments 1 (NSYR) and 2 (YSYR) were gathered as exhibited in Tables 4 and 5.

Table 4. Demographic Statistics of Experiment 1 (NSYR)

Variable/Facotrs	Frequency	Percentage
Gender		
Male	27	25
Female	81	75
Age		
18-20	13	12
21-23	86	79..6
24-26	9	8.3
Program		
Bsc Information Technology	55	50.9
Bsc Computer Science	2	1.9
Bsc Multimedia	47	43.5
Bsc Education	1	0.9
Bsc Business Mathematics	2	1.9
Bsc Network	1	0.9
Course Subjects		
Database	63	58.3
Introduction to Programming Java	28	25.9
System Analysis and Design	3	2.8
Basic Programming	5	4.6
Expert System	3	2.8
Software Engineering	2	1.9
Artificial Intelligence	2	1.9
Basic Networking	2	1.9
Semester		
Semester 1	5	4.6
Semester 2	24	22.2
Semester 3	20	18.5
Semester 4	54	50.5
Semester 6	5	4.6

Table 5. Demographic Statistics of Experiment 2 (YSYR)

Variable/Facotrs	Frequency	Percentage
Gender		
Male	8	34.8
Female	15	65.2
Age		
18-20	6	26.1
21-23	14	60.9
24-26	3	13.0
Program		
Bsc Information Technology	18	78.3
Bsc Multimedia	5	21.7
Course Subjects		
Database	9	39.1
Programming Enhancement Program	11	47.8
Basic Programming	1	4.3
Expert System	1	4.3
Basic Networking	1	4.3
Semester		
Semester 2	4	17.4
Semester 3	5	21.7
Semester 4	10	43.5
Semester 6	3	13.0
Semester 9	1	4.3

B. Structural Equation Modeling

SEM is a methodological technique to ease the analytical complex model. Further, it is a statistical technique for addressing a confirmatory approach of a structural theory that generates observation on multiple variables [50], [51]. Research has shown that there are two types of SEM named as the Covariance-Based SEM (CB-SEM) and Partial Least Square SEM (PLS-SEM). The CB-SEM is purposely designed for estimating the parameters of the model in order to reduce the variation between the sample covariance and those predicted by the theoretical model. It reduces the efforts to predict the existence of dependent variables through the maximization of the variance explained (R2) of the dependent variable [52]. In contrast, PLS-SEM is capable of making use of both normal and non-normal dataset. Hence, this study uses PLS-SEM to analyze the collected data.

C. Analytical Activities in Structural Equation Modeling

The assessment of PLS-SEM covers two different approaches specifically for achieving different objectives, which are measurement model and structural model assessments [53]. The first approach is known as the measurement model evaluation, which addresses the reliability and validity of measures that form embedded constructs [53], [46]. In detail, Hair et al. [54] and Chin [46] emphasizes that major activities in evaluating the measurement model are internal consistency reliability, indicator reliability, convergent validity, and discriminant validity [55]. Besides that, structural model analytical phase in SEM also addresses the significance of the path coefficients and level of R2 [54], [46].

1) Reliability of Internal Consistency

Within PLS, composite reliability (CR) is used to measure the internal consistency [56]. CR takes into consideration the difference in loadings of the indicators [57]. The reliability of an internal consistency is deemed satisfactory when the value is at the minimum level (0.7) in the early stage of research and increases to 0.8 or 0.9 in the later stages. Meanwhile, any value below 0.6 reflects a lack of reliability [58]. For this study, the CR for each construct is shown in Tables 6 and 7, which are greater than 0.7. This indicates that the internal consistency is satisfactory.

Table 6. Descriptive and Reliability Statistics for NSYR

Construct	Items	Means	Standard Devision	Loading	T-Statistic	CR	AVE
Sociallization	SF1	4.00	0.820	0.9455	2.8634		0.5796
	SF2	4.18	0.818	0.7164	2.4294		
	SF3	4.23	0.793	0.5371	1.5798	0.8697	
	SF4	4.32	0.734	0.7149	2.4504		
	SF5	4.12	0.872	0.8312	2.9089		
Externalization	E1	3.634	1.010	0.9039	2.0797		0.5
	E2	4.09	0.803	0.5605	1.3762	0.741	
	E4	3.87	0.928	0.6071	1.4378		
Combination	C2	4.08	0.866	0.4411	1.2179		0.577
	C4	3.40	1.160	0.8466	2.6186	0.7912	
	C5	3.57	1.070	0.9053	2.6838		
Internalization	IIODMI1	3.65	0.889	0.888	3.1684		0.5106
	IIODMI2	3.09	0.981	0.6358	1.9121		
	IIODMI3	3.74	0.741	0.8341	3.1045		
	IIODMI5	3.58	0.844	0.7677	2.979	0.8767	
	IIOT3	3.95	0.847	0.5645	1.6693		
	IIOT4	3.94	0.795	0.65775	2.2835		
	IIOT5	3.93	0.817	0.5877	1.831		
Code Quality	NSYR	4.50	1.204	1	0		1

Table 7. Descriptive and Reliability Statistics for YSYR

Construct	Items	Means	Standard Devision	Loading	T-Statistic	CR	AVE
Sociallization	SF1	4.13	0.920	0.5775	1.8629	0.9186	0.6982
	SF2	4.30	0.703	0.9478	3.9258		
	SF3	4.35	0.714	0.9201	3.3777		
	SF4	4.30	0.703	0.8562	3.0366		
	SF5	3.91	1.083	0.8243	3.7991		
Externalization	E2	4.17	0.778	0.5145	1.207	0.7682	0.5182
	E4	3.70	1.105	0.8784	2.297		
Combination	C4	3.52	1.238	0.1697	0.3843	0.7805	0.5097
	C5	3.43	1.119	0.9952	4.5707		
Internalization	IIODMI1	3.78	0.902	0.8559	3.2322	0.923	0.5245
	IIODMI2	4.09	0.733	0.6017	2.1824		
	IIODMI3	3.43	1.161	0.7347	2.6849		

Construct	Items	Means	Standard Devision	Loading	T-Statistic	CR	AVE
	IIODMI5	4.00	0.739	0.6149	1.982		
	IIOT3	4.04	0.767	0.7181	2.1625		
	IIOT4	3.91	0.900	0.753	2.202		
	IIOT5	3.96	0.767	0.438	2.6714		
	IIOL2	3.17	0.885	0.6688	2.2139		
	IIOL3	4.04	1.054	0.6232	2.2286		
	IIOL5	3.74	1.114	0.8432	3.1003		
	IIOL7	3.65	0.878	0.7581	2.4053		
Code Quality (YSYR)	YSYR	3.57	2.233	1	0		1

2) Indicator Reliability

In order to assess indicators' reliability, this study needs to evaluate the extent a variable or a group of variables is proportionate with what it means to measure [49]. Accordingly, the reliability construct was evaluated independently. With reference to Urbach and Ahlemann [49], indicator loadings must be significant at minimum 0.05 and the loading should be greater than 0.7. This is because with the loading value at 0.7, a latent variable (LV) is considered to be able to explain at least 50 percent of its indicator's variance. On the other hand, Bootstrapping is resampling method that can be used to examine the significance of the indicator loadings. In general, the decision of eliminating an indicator should be taken carefully when considering PLS characteristics of consistency [59]. In case of low value of an indicator, it is logic to take the decision of eliminating that indicator and that elimination is linked with the significant increase of CR value [57]. Therefore, the indicator reliability in NSYR model ranges from 0.741 to 0.8767 as shown in Table 6 and in YSYR model, the indicator reliability ranges from 0.7682 to 0.923 as shown in Table 7.

3) Convergent validity

Convergent validity indicates the extent to which individual items reflect a construct converging as compared with items that measure various constructs [49].

With the aid of PLS, the value of average variance extracted (AVE) is used to calculate the convergent validity. According to Fornell and Larcker [60], in case of AVE value of a construct amount is not less than 0.5, then the convergent validity is considered sufficient.

In regards to that, the convergent reliability for NSYR model for this study is exhibited in Table 6. It reveals that the entire construct AVE values are above the threshold value (0.5). In the context of this research, the AVE ranges from 0.5 to 0.5796. This shows that the analysis satisfies the AVE rule.

Further, the CR for YSYR model is shown in Table 7 and reveals that the entire construct AVE values are above the threshold value (0.5). In the context of this research, the AVE is ranges from 0.5097 to 0.6982. This also satisfies the AVE rule.

4) Discriminant validity

Discriminant validity is used to distinguish one measure from another of a construct measures. On the contrary, the convergent validity, discriminant validity examines whether the items intentionally measure another issue [49]. Within PLS, cross loading [56] and standard of Fornell-Larcker [60] are two commonly used measures of discriminant validity. The first measurement analysis was conducted by examining the AVE for both YSYR and NSYR models and represented in Tables 8 and 9.

Table 8. Discriminant Validity for YSYR

	Code Quality (YSYR)	Combination	Externalization	Internalization	Socialization
Code Quality (YSYR)	**1**	0	0	0	0
Combination	-0.4662	**0.7139**	0	0	0
Externalization	-0.2896	0.5192	**0.7199**	0	0
Internalization	0.2918	0.414	0.3571	**0.7242**	0
Socialization	-0.2548	0.3401	0.4254	0.6092	**0.8356**

Table 9. Discriminant Validity for NSYR

	Code Quality (NSYR)	Combination	Externalization	Internalization	Socialization
Code Quality (NSYR)	**1**	0	0	0	0
Combination	-0.2217	**0.7596**	0	0	0
Externalization	0.0904	0.3718	**0.7071**	0	0
Internalization	0.2378	0.1966	0.4487	**0.7146**	0
Socialization	0.1049	0.3022	0.6308	0.4983	**0.7613**

D) Validation of Structural Model

Validation of the structural model can assist this study to systematically estimate whether the data support the hypotheses characterized by the structural model [49]. It is not proper to establish the analysis of the structural model unless the measurement model has been achieved successfully. Within PLS, a coefficient of determination (R2), and path coefficients are used to evaluate the structural model.

1) Coefficient of Determination (R^2)

The variance explanation of R^2 measures the relationship of latent variables to its total variance. Based on the benchmark by Chin [46], R^2 is considered weak if it is 0.19 and below. R2 of 0.333 is accepted as the average, while R2 of 0.67 is considered as substantial.

Figure 4 and 5 represent the results of structural model for NSYR and YSYR obtained in this study respectively.

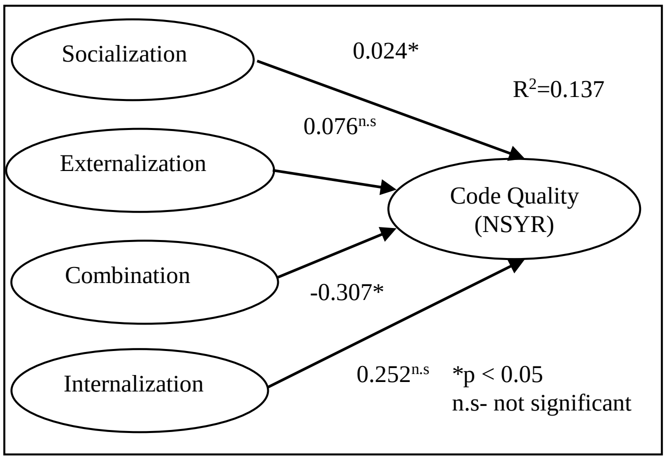

Figure 4. Result of NSYR structural model

With reference to Figure 4, Socialization, Externalization, Combination, and Internalization are able to explain 13.7% of the variance in code quality of NSYR. This shows that coefficient of determination R^2 is weak. On the other hand, Figure 5 reveals that Socialization, Externalization, Combination, and Internalization are able to explain 72.4% of the variance on code quality of YSYR.

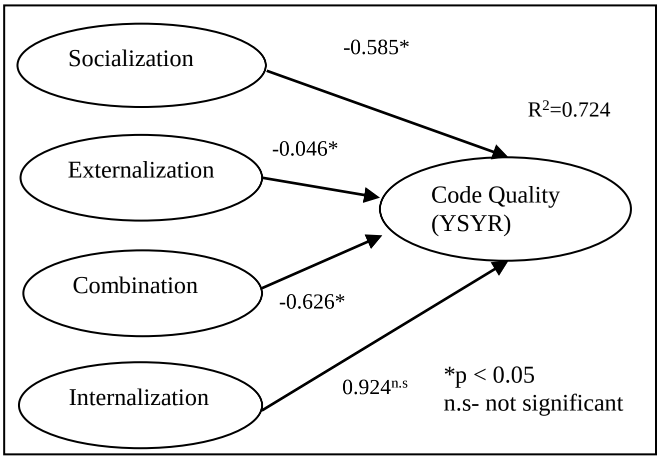

Figure 5. Result of YSYR structural model

2) Path Coefficient

By testing the path coefficient value, this study is eligible to know whether the relationship between two LV is strong enough. In order to investigate the relationship between two LVs, this study needs to notice the path coefficients, algebraic sign, magnitude, and significance. According to Huber et al. [61], the impact of the model would be felt if the path coefficient is greater than 0.100 and significant to support the hypothesis at 0.05 significant level.

Having run the test, the results are shown in Table 10. It could be seen that T-test values help this study to judge which of the hypothesis are supported. When T-test is above or equal 0.9, the hypothesis is supported, otherwise the hypothesis is not supported [46].

Table 10. Standard Values for Assessing Measurement Model

Dependent Variables	Independent Variable	Path Coefficient ()	Observed T-statistics	Signifi-cant Level
Code Quality (YSYR)	Socialization	-0.5852	3.5097	0.05
	Externalization	-0.0456	0.3025	0.05
	Combination	-0.6258	2.0617	0.05
	Internalization	0.9236	2.4107	0.05
Code Quality (NSYR)	Socialization	0.0241	2.0776	0.05
	Externalization	0.0762	0.2164	0.05
	Combination	-0.3069	2.8001	0.05
	Internalization	0.2519	1.6609	0.05

Further, Table 11, Figure 6 and Figure 7 show the supported hypotheses for this study based on the results in Table 10.

Table 11. Supported Standard Hypotheses of The Study

Hypothesis	Codification	Description	Result
H1	$S \xrightarrow{NS} CQ$	The Socialization process contributes positively to student's code quality without employing SECI process.	Supported
H2	$E \xrightarrow{NS} CQ$	The Externa-lization process contributes positively to student's code quality without employing SECI process.	Not Supported
H3	$C \xrightarrow{NS} CQ$	The Combination process contributes	Supported

Hypothesis	Codification	Description	Result
		positively to student's code quality without employing SECI process.	
H4	I $\xrightarrow{NS}$ CQ	The Interna-lization process contributes positively to student's code quality without employing SECI process.	Not Supported
H5	S $\xrightarrow{YS}$ CQ	The Socialization process contributes positively to student's code quality with employing SECI process.	Supported
H6	E $\xrightarrow{YS}$ CQ	The Externa-lization process contributes positively to student's code quality with employing SECI process.	Supported
H7	C $\xrightarrow{YS}$ CQ	The Combination process contributes positively to student's code quality with employing SECI process.	Supported
N8	I $\xrightarrow{YS}$ CQ	The Internalization process contributes positively to student's code quality with employing SECI process.	Not Supported
H9	SECI $\rightarrow$ CQ	SECI process contributes positively to student's code quality.	Supported

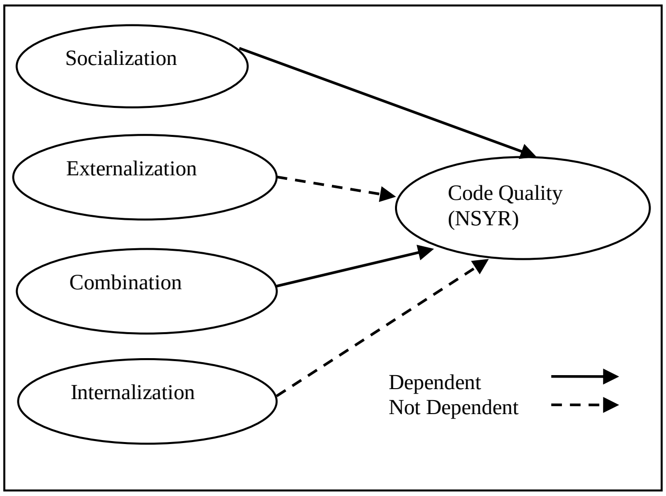

Figure 6. Results of the hypothesis for experiment 1 (NSYR)

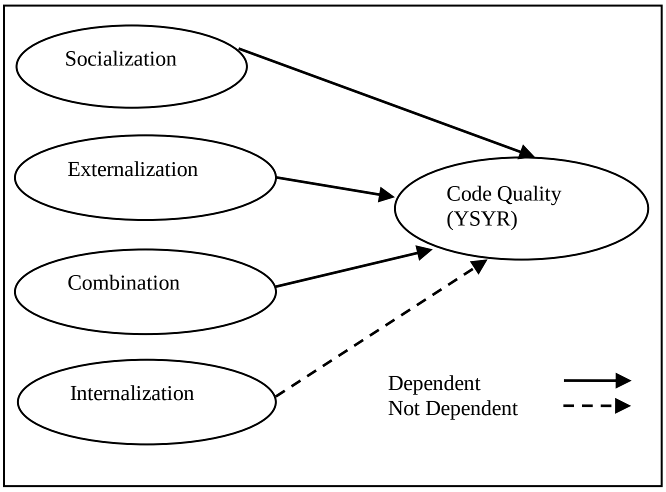

Figure 7. Results of the hypothesis for experiment 2 (YSYR)

V. Discussion and Conclusion

The experiments were divided into two groups with the dependent variable of code quality of the first experiment is named NSYR and the dependent variable of code quality of the second experiment is tagged as YSYR. Meanwhile, the independent variables are uniform for both groups as socialization (SC), Externalization (EXT), Combination (CMB), Internalization (INT). The objective of this study concerns on investigating the relationships between each of the four processes of SECI model and code quality.

The effect of socialization on the code quality is generally believed that interaction or sharing of knowledge in a virtual way or from tacit to tacit form may not yield full understanding to the listeners or pair groups based on the individual intelligence level. The literatures reveal that there is a relationship between the sharing of knowledge in the form of tacit to tacit between two people or groups towards achieving a code quality. This is confirmed by the results of the two experiments with model and without model (YSYR and NSYR). The implication of this result in the student without model (NSYR) is that participants have found to have prior or basic knowledge on Java programming language.

This enables them to transfer the knowledge between PP members without documentation and achieve code quality. In the context of PP laboratory assignment (YSYR), the results show that it would be easier for the participants to achieve code quality. This is as a result of their exposure to the knowledge of Java programming language. These explain that socialization is significantly related to code quality and it is in line with the study Singh and friends [36]. Meanwhile, the relationship between the Socialization process and code quality by the participants with model (YSYR) is better than the relationship between Socialization and code quality by the participant without model (NSYR) (t values = 3.5097 and 2.0776 respectively).

The results reveal that there is no significant relationship between the driver and the navigator in the effect of externalization on code quality. This is based on the obtained results from the analysis of the collected data in the two experiments (NSYR and YSYR). The obtained result is consistent with the study by Ahmad et al. [19], which affirms that achieving a project's completion (the transfer of knowledge from abstract to documented form) does not bring any improvement on the code quality. Additionally, the results of the hypotheses may hint a lack of code drafting before simply writing the codes in the computer. On the other hand, the Externalization towards code quality in NSYR and YSYR ended not significant. However, the result of YSYR is better than NSYR with t value = 0.3025 and 0.2164 respectively.

The effect of combination on code quality is one of the knowledge management models, which focuses on sharing or transferring of knowledge between the pair from explicit format to explicit format. The obtained results in both experiments 1 and 2 (YSYR and NSYR), shown in Table 11 support the statement that the relationship between Combination and code quality is significant. This means that it is mandatory to document the references that guide the code quality could be achieved through the Combination form knowledge transfer. Hence, the obtained result is consistent with the previous study by Ahmad et al. [19]. This implicates that it provides people who are involved in the learning and sharing of programming skill to develop quality code, should they have access to references while writing the codes for the given assignment. Besides that, the comparison between the two programming assignments shows that assignment without model is better than assignment with model. This is deduced based on the t value = 2.8001 and 2.0617 respectively.

Meanwhile, the effect of internalization on code quality in the SECI model is described as systematic explicit knowledge, which can be converted into a richer consistent and more complicated tacit knowledge, such as saved in human memory (memorization). It was initially hypothesized that there is a significant relationship between knowledge shared from concrete to an abstract form when determining or seeking for programming skills. In the context of this research, both analyses in the two experiments (with model and without model) confirm that there are significant relationships between Internalization and code quality of Java programming assignment. The obtained findings are in line with the previous studies that support the hypothesized statement [43]. This implies that the exchange of knowledge from explicit form to tacit form while addressing Java programming language helps in achieving code quality. Finally, the comparison of results of the two experiments show that YSYR is better than NSYR through t values = 2.4107 and 1.6609 respectively.

Conclusively, the significant findings among the four research hypotheses show that only one construct is agreeably not supported in the two experiments, which is Externalization. In contrast, Socialization in YSYR is found as the most influential factor among the SECI processes.

This study has contributed in providing a road map for the educators to achieve code quality using effective teaching methods through determining the impact factors for determining PP knowledge-based sharing for improving programming skills. Above of that, this study provides the empirical evidence on the impact of each Socialization, Combination and Internalization on code quality.

As it was stated that this research provides the stakeholders at higher learning institution, the needs to achieve effective program code quality. Consequently, the number of participants needs to be increased in the future research in order to achieve robust results and also the qualitative research approach should be added to the work in order to obtain full representative of the participants' mind.

The importance of achieving a code quality while dealing with PP knowledge sharing at higher learning institution cannot be overemphasized. Therefore, this calls for immediate recommendation of this research at higher learning institution since the research has identified the influential factors for achieving program code quality and knowledge sharing PP.

Acknowledgment

The authors wish to thank the Ministry of Education Malaysia for funding this study under Fundamental Research Grant Scheme (FRGS-SO/CODE:12814).

References

[1] E. di Bella, I. Fronza, N. Phaphoom, A. Sillitti, G. Succi, J. Vlasenko, "Pair Programming and Software Defects-A Large, Industrial Case Study". *IEEE Transactions on Software Engineering*, 39 (7), pp. 930-953, 2013.

[2] A. Sillitti, G. Succi, J. Vlasenko. "Understanding the impact of pair programming on developers attention*", In Proceedings of the 2012 International Conference on Software Engineering*, Zurich, pp. 1094-1101, 2012.

[3] S. Xu, V. Rajlich. "Empirical validation of test-driven pair programming in game development". In *Computer and Information Science, 2006 and 2006 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse. ICIS-COMSAR 2006. 5th IEEE/ACIS International Conference on*, pp. 500-505, 2006.

[4] M. Omar, S.-L. Syed-Abdullah, A. Yasin. "Adopting Agile Approach: A Case in Malaysia". In *Proceedings* KMICe, pp.544-549, 2010.

[5] S. L. Syed-Abdullah, M. Omar, M. N. A. Hamid, C. L. bt Ismail, K. Jusoff. "Positive affects inducer on software quality", *Computer and Information Science*, 2 (3), pp. 64-70, 2009.
[6] K. B. Rimington. "Expanding the Horizons of Educational Pair Programming: A Methodological Review of Pair Programming in Computer Science Education Research". *A Thesis Submitted for Master of Science (MS),* UtahState University, 2010.
[7] W. Chigona, M. Pollock. "Pair programming for information systems students new to programming: Students' experiences and teachers' challenges". *in Management of Engineering & Technology, 2008. PICMET 2008. Portland International Conference on*, pp. 1587-1594, 2008.
[8] Md. Rejab, M. Omar, M Ahmad. "Knowledge Internalization in Pair Programming Practices", *Journal of Information and Communication Technology (JICT),* 11, pp. 163-177, 2013.
[9] V. Venkatesan, A. Sankar. "Investigation of Student's Personality on Pair Programming to Enhance the Learning Activity in the Academia", *Journal of Computer Science*, 10 (10), pp. 2020-2028, 2014.
[10] I. Nonaka, H. Takeuchi. "The knowledge-creating company", *Harvard Business Review*, pp. 96-103, 2000.
[11] L. Porter, M. Guzdial, C. McDowell, B. Simon. "Success in introductory programming: what works?", *Communications of the ACM*, 56 (8), pp. 34-36, 2013.
[12] L. Williams. "Lessons learned from seven years of pair programming at North Carolina State University", *ACM SIGCSE Bulletin*, 39 (4), pp. 79-83, 2007.
[13] A. S. M. Sajeev, S. Datta. "Introducing Programmers to Pair Programming: A Controlled Experiment", *Agile Processes in Software Engineering and Extreme Programming Lecture Notes in Business Information Processing*, 149, pp. 31-45, 2013.
[14] S. Wray. "How pair programming really works". *IEEE software*, 27, pp. 50-55, 2010.
[15] J. Yi. "Externalization of tacit knowledge in online environments", *International Journal on E-learning*, 5 (4), pp. 663-674, 2006.
[16] M. Polanyi. *The study ofman*, Routledge &amp, London, 1958.
[17] S. Guthrie. "The Role of Tacit Knowledge in Judgement and Decision Making". *In Proceedings of the 1995 International Conference on Outdoor Recreation and Education*, pp.104-115, 1996.
[18] R. K. Kavitha, M. I. Ahmed. "Knowledge Sharing Through Pair Programming in Learning Environments: An empirical study". *In: Education and Information Technologies*. Springer, US, 20, pp. 319-333, 2013.
[19] M. Ahmad, M. R. Mawarny, M. S. Abdulah, O. Mazni, B. A. Khairul, A. Merza. "Measuring Tacit Knowledge Acquired During Problem Based Learning Teaching Method in Learning Management System Environment". *To appear in Procedia Computer Science, Elsevier B. V,* 1, pp. 775-781, 2012.
[20] S. Billett, S. Choy. "Learning through work: emerging perspectives and new challenges", *Journal of Workplace Learning*, 25 (4), pp. 264-276, 2013.
[21] P. Dudley. "Teacher learning in Lesson Study: What interaction-level discourse analysis revealed about how teachers utilised imagination, tacit knowledge of teaching and fresh evidence of pupils learning, to develop practice knowledge and so enhance their pupils' learning", *Teaching and teacher education*, 34, pp. 107-121, 2013.
[22] S. Ryan, R. V. O'Connor. "Acquiring and sharing tacit knowledge in software development teams: An empirical study", *Information and Software Technology*, 55 (9), pp. 1614-1624, 2013.
[23] J. Wan, D. Wan, W. Luo, X. Wan. "Research on Explicit and Tacit Knowledge Interaction in Software Process Improvement Project", *Journal of Software Engineering and Applications*, 4 (6), p. 335-344, 2011.
[24] G. Murphy, S. Salomone. "Using social media to facilitate knowledge transfer in complex engineering environments: a primer for educators". *European Journal of Engineering Education*, 38 (1), pp. 70-84, 2013.
[25] N. J. Foss, J. Lyngsie, S. A. Zahra. "The role of external knowledge sources and organizational design in the process of opportunity exploitation". *Strategic Management Journal*, 34, pp. 1453-1471, 2013.
[26] W. He, V. Cho, C. Qi, X. Xu, F. Lu. "Linking Knowledge Sharing and Employee Creativity: Decomposing Knowledge Mode and Improving the Measure of Tacit Knowledge Sharing", In *Pacific Asia Conference on Information Systems (PACIS 2013) Proceedings*, pp. 219, 2013.
[27] C. Kimble. "What cost knowledge management? The example of Infosys", *Global Business and Organizational Excellence*, 32 (3), pp. 6-14, 2013.
[28] W. R. King. *Knowledge management and organizational learning*, Springer, 2009.
[29] F. Alipour, K. Idris, R. Karimi. "Knowledge creation and transfer: role of learning organization", *International Journal of Business Administration*, 2 (3), pp. 61-67, 2011.
[30] U. Makambe. "Perspectives on Knowledge Management - A Literature Review", *Information and Knowledge Management*, 5, (1), pp. 88-97, 2015.
[31] L. Aggestam. "Learning Organization or Knowledge Management – Which Came First, The Chicken or The Egg?", *Information Technology and Control*, 35 (3A), pp. 295-302, 2006.
[32] M. A. Jabar, F. Sidi, M. H. Selamat. "Tacit knowledge codification", *Journal of Computer Science*, 6 (10), pp. 1170-1176, 2010.
[33] S. Y. Sung, J. N. Choi. "Do organizations spend wisely on employees? Effects of training and development investments on learning and innovation in organizations", *Journal of Organizational Behavior*, 35 (3), pp. 393-412, 2014.
[34] B. Crawford, C. L. de la Barra, R. Soto, M. Dorochesi, E. Monfroy. "The Role of Knowledge Management in Agile Software Development". In *HCI International 2013-Posters' Extended Abstracts, ed: Springer*, 373, pp. 17-21, 2013.
[35] D. Sharma. "Knowledge Management & Organizational Structure: A Study Of Indian Companies", *International Journal of Organizational Behaviour & Management Perspectives,* 2 (4), pp. 585-593, 2014.

[36] A. Singh, K. Singh, N. Sharma. "Knowledge Management: the agile way". In *Information and Knowledge Management*, pp. 143-152, 2013.

[37] M. M. Rejab, M. Omar, B. A. Khairul, M Ahmad. "Pair Programming In Inducing Knowledge Sharing". In *Proceedings of the 3rd International Conference on Computing and Informatics, ICOCI*, pp. 251-258, 2011.

[38] A. Fengjie, Q. Fei, C. Xin. “Knowledge sharing and web-based knowledge-sharing platform”. In *Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business*, pp. 278-281, 2004.

[39] M. Omar, R. Romli, A. Hussain. "Automated Tool to Assess Pair Programming Program Quality", *In Proceedings KMICe*, 2008.

[40] N. Salleh, E. Mendes, J. Grundy. "Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review", *Software Engineering, IEEE Transactions on*, 37, pp. 509-525, 2011.

[41] E. C. K. Cheng. “Knowledge Management for School Development”, *SpringerBriefs in Education*, VII (83), pp. 11-23, 2015.

[42] I. Nonaka, H. Takeuchi. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*, Long Range Planning, OUP USA, 1995.

[43] C. Cayaba, Z. Pablo. "A Qualitative Investigation of the SECI Model’s Knowledge Conversions in the Applications Development Context", *In Pacific Asia Conference on Information Systems (PACIS 2013) Proceedings*, pp. 265, 2013.

[44] H. Erdogmus. "On the effectiveness of test-first approach to programming", *Software Engineering, IEEE Transactions on*, 31 (3), pp. 226 -237, 2005.

[45] C. Wohlin, P. Runeson, M. Hst , M. C. Ohlsson, B. Regnell, A. Wessln. *Experimentation in software engineering*, Springer Publishing Company, Incorporated, 2012.

[46] W. W. Chin. *The partial least squares approach to structural equation modeling, In G. A. Marcoulides (Ed.), Modern Methods for Business Research*, Lawrence Erlbaum Associates, Mahwah, 2010.

[47] R. B. Kline, *Principles and Practice of Structural Equation Modeling*, The Guilford Press, New York, 2011.

[48] C. Fornell, F. L. Bookstein. "Two structural equation models: LISREL and PLS applied to consumer exit-voice theory", *Journal of Marketing research*, 19 (4), pp. 440-452, 1982.

[49] N. Urbach, F. Ahlemann. "Structural equation modeling in information systems research using partial least squares", *Journal of Information Technology Theory and Application*, 11, pp. 5-40, 2010.

[50] P. M. Bentler. *EQS Structural Equations Program Manual*, BMDP Statistical Software, Los Angeles, 1989.

[51] B. M. Byrne. *Structural Equation Modeling with AMOS: Basic Concepts, Applications, and Programming*, Taylor & Francis Group, NY, 2010.

[52] C. Barroso, G. C. Carrión, J. L. Roldán. *Applying maximum likelihood and PLS on different sample sizes: studies on Servqual model and employee behavior model*, In:Vinci,V.E.,Chin,W.W.,Henseler,J.,Wang,H.(Eds.),Handbook of Partial Least Squares: Concepts, Methods and Applications, Springer-Verlag,Berlin, Germany, 2010.

[53] B. Wilson. *Using PLS to investigate interaction effects between higher order brand constructs. In Handbook of Partial Least Squares: Concepts, Methods, and Applications”,* (Esposito Vinzi V, Chin Ww, Henseler J and Wang H, Eds), vol. 2 of Computational Statistics, Springer, Heidelberg, 2010.

[54] J. F. Hair, R. E. Anderson, R. L. Tatham, W. C. Black. *Multivariate data analysis*, 7th edition, Pearson New International Edition, NJ, 2013.

[55] B. R. Lewis, G. F. Templeton, T. A. Byrd. "A methodology for construct development in MIS research", *European Journal of Information Systems*, 14 (4), pp. 388-400, 2005.

[56] D. Straub, M.-C. Boudreau, D. Gefen. "Validation guidelines for IS positivist research", *The Communications of the Association for Information Systems*, 13, p. 63, 2004.

[57] K. F. Hashim. "Understanding the Determinants of Continuous Knowledge Sharing Intention Within Business Online Communities”. *A Thesis Submitted for the Degree of Doctor of Philosophy (PhD),* Auckland University of Technology, AUT University, 2012.

[58] I. H. Bernstein, J. Nunnally, *Psychometric theory*, McGraw-Hill, New York, 1994.

[59] J. Henseler, C. M. Ringle, R. R. Sinkovics. "The use of partial least squares path modeling in international marketing", *Advances in international marketing*, 20, pp. 277-319, 2009.

[60] C. Fornell, D. F. Larcker. "Evaluating structural equation models with unobservable variables and measurement error", *Journal of marketing research*, 18 (1), pp. 39-50, 1981.

[61] F. Huber, A. Herrman, F. Meyer, J. Vogel, K. Vollhard. *Kausal modellierung mit Partial Least Squares*, Eine anwendungsorientierte Einfuehrung , first ed. Gabler, 2007.

Author Biographies

Mazida Ahmad is a senior lecturer at the School of Computing, College of Arts and Sciences, Universiti Utara Malaysia. She received the BMIS degree from International Islamic University of Malaysia, in 2001, the MSc. degree in Software Engineering from Universiti Teknologi Malaysia, in 2003, and the Ph.D. degree in Knowledge Management from Universiti Sains Malaysia, in 2010. Her current research interests include knowledge management, information system development and software engineering education.

Ariffin Abdul Mutalib is a senior lecturer at the School of Multimedia Technology and Communication, College of Arts and Sciences, Universiti Utara Malaysia. He received the Bachelor of Information Technology from Universiti Utara Malaysia in 1999, MSc. in Interactive Multimedia (HCI) from Heriot-Watt University, Scotland, in 2001, and Ph.D. in Information Technology from Universiti Utara Malaysia, in 2009. His current research interests include Human-computer Interaction, Usability, and Computers in teaching and learning. Among projects he has ventured into include interactive contents for various groups of children including those with special needs.

Mazni Omar is a senior lecturer at the School of Computing, College of Arts and Sciences, Universiti Utara Malaysia. She received the BSc. degree (with honors) in information technology from Universiti Utara Malaysia, in 2000, the MSc. degree in software engineering from Universiti Teknologi Malaysia, in 2002, and the Ph.D. degree in information technology and quantitative sciences from Universiti Teknologi MARA, Malaysia, in 2012. Her current research interests include agile software development, empirical software engineering, software quality and data mining.

Azman Yasin is an Associate Professor at the School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, Sintok, Kedah. Malaysia. He holds PhD in Information Technology from Universiti Kebangsaan Malaysia. His research interest includes software engineering, information retrieval specifically scheduling and timetabling using artificial intelligence techniques.

Rohaida Romli is currently working as a lecturer at School of Computing, Universiti Utara Malaysia since the year of 2000. Her area of specialization is Software Engineering. In terms of the academic background, she did her first and master degrees (in 1998 and 2000 respectively) both at Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia in the fields of Industrial Computing and Computer Science respectively. She has been awarded with the PhD in Software Engineering from Universiti Sains Malaysia at the end of 2014. In terms of the current research interest, she is more focusing on researches with regard to software engineering, software verification and testing, and automated programming assessment in computer science education.

.